Session 117

Image Capture Framework

Travis Brown
Mac OS X Imaging Technology Manager

Digital Photography Is Booming!

- According to InfoTech Trends, a technology research company:
  - Digital camera shipments will grow at approximately 26% in the US alone 2000–2001
  - Worldwide, the Average Growth Rate jumps to about 43% 2000–2004
  - This year, digital cameras will account for 36% of all cameras sold, both film and digital
  - In 2003, that number is expected to grow to 62%

Sources: Electronic News (02/05/01), Office Solutions (01/01/01), Electronic Buyers' News (07/24/00), Electronic News (02/14/00)

Session 117

Image Capture Framework

Steve Swen
Color Imaging Engineering Manager

Session Schedule

- Introduction
- Architecture Overview
- Application Level APIs
- Device Level Support

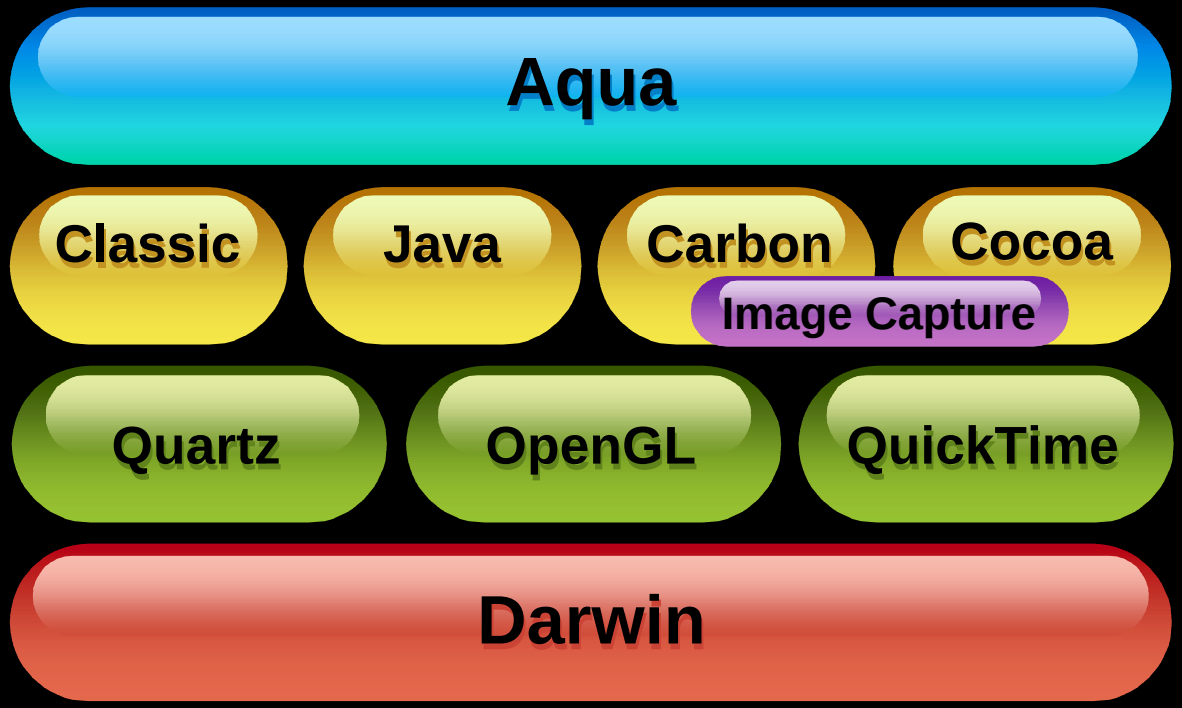
Imaging on Mac OS 9

- Mac OS 9 has no system-level input architecture
- Most devices ship with proprietary solutions/drivers
- Applications could not work with image capture devices directly

Image Capture on Mac OS X

- Image Capture Framework is a new API collection and set of system services
- Gives applications easy access to image capture devices such as digital still cameras
- System-level integration to leverage Apple technologies

What Is Image Capture?



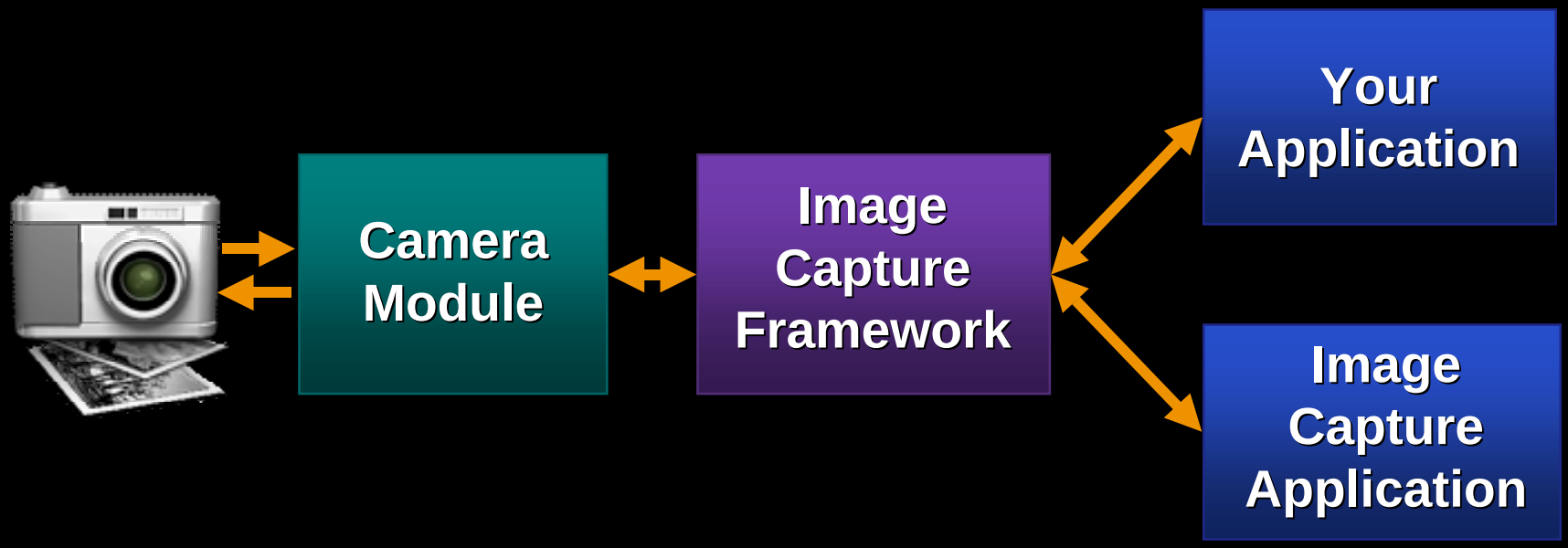
Session Schedule

- Introduction
- Architecture Overview
- Application Level APIs
- Device Level Support

What's in Image Capture Framework

- Framework and APIs for applications and imaging devices
- Built-in support for cameras using standard protocols
- Image Capture Application to preview and download images

Image Capture Framework



Built-in Camera Support

- PTP (Picture Transfer Protocol) class
- Digita® class
- Mass storage class
- Removable media

PTP Class (aka. PIMA 15740)

- Industry-standard camera protocol defined by platform vendors and device manufacturers
- Protocol enables full control of camera functions
- Transport independent
- Mac OS X 10.0 supports a subset of PTP (“plug and drain” operations)
- Recent cameras from Kodak and Sony support PTP protocol

Digita® Class

- Vendor-specific protocol by Flash Point Technology
- Some Kodak and HP cameras use this protocol
- Let us know if you use this protocol

Mass Storage Class

- Most common camera protocol
- Mounts camera as MS-DOS file system
- Cannot control camera or get camera specific properties

Removable Media

- CompactFlash™, SmartMedia™, Memory Stick™, Microdrive™…
- Floppy disk, SuperDisk™, Recordable CD…
- Other storage devices mount as file systems

Recognizing Storage Device

- Image Capture intercepts volume mounts
  - Mount as MS-DOS file system
  - Has standard DCF file structure
  - Registered custom folders and file structure
  - Send us your file structure if you store image files outside the DCIM folder

Supported File Formats

- Still image file formats
  - JPEG, TIFF, GIF
- Video file formats
  - QuickTime, MPEG
- Audio file formats
  - MP3
- Let us know if you support file types not on the list

Attention Camera Vendors

- Use standard protocols: PTP and Mass Storage
- Send prototypes to us to ensure compatibility
- Send custom file structure and file types to us

Image Capture Application

- Light-weight application to browse and download images
- Hot plug actions and automatic task to customize users workflow
- Scripts to do common tasks

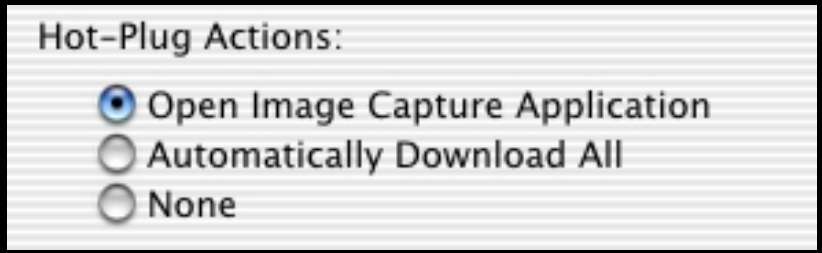
Image Capture Application

Download Synchronization

- Avoid unnecessary downloads
- Mac OS X 10.0 uses filename and size
- Will use filename and capture date in meta data in future release

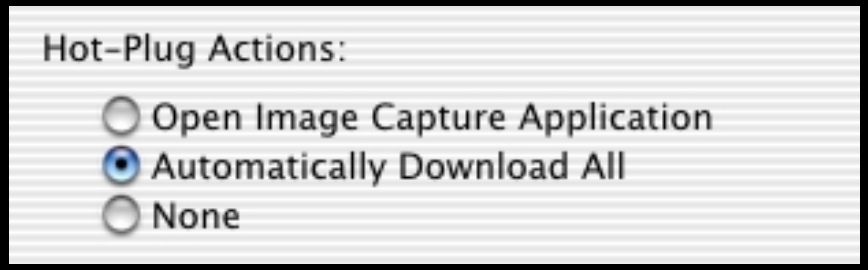
Hot Plug Actions

- Open Image Capture Application
  - User can browse and download selected images



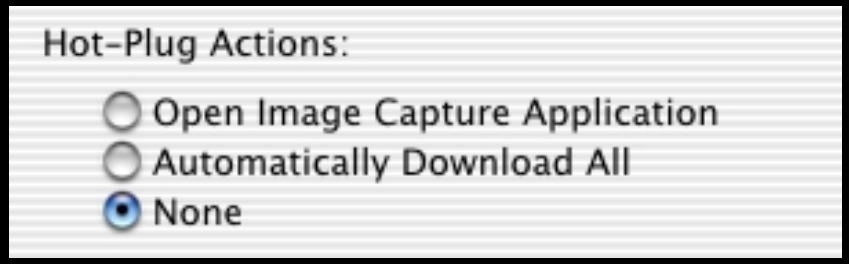
Hot Plug Actions

- Automatically download all
  - Download all files and launch selected automatic task if set
  - Pass the downloaded files to the application



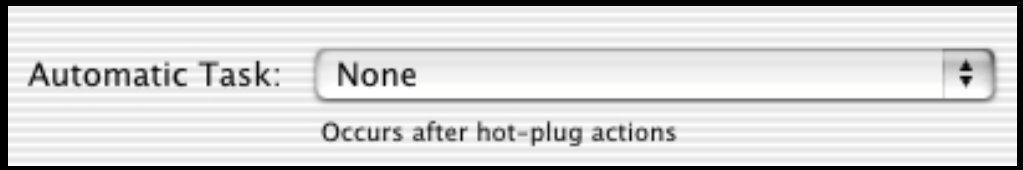
Hot Plug Actions

- None
  - Does nothing in Mac OS X 10.0
  - User can launch a selected application in a future version



Automatic Task

- Can be AppleScripts, applications, or aliases
- Search path includes
  - /Library/Image Capture/Scripts



Future Directions for the Image Capture Application

- Meta data browser
- Per image rotation/delete
- Preference panel

Scanner Support on Mac OS X

- Image Capture will support scanners in the future
- If you need to support scanners in the near term:
  - Port your Mac OS 9 scanner application to Carbon
  - Support your scanner device using TWAIN

Demo

Image Capture Tour

Session Schedule

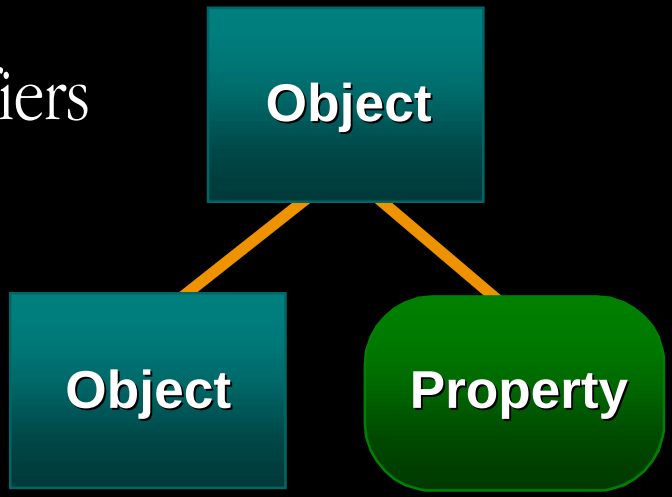
- Introduction
- Architecture Overview
- Application Level APIs
- Device Level Support

Application Level APIs

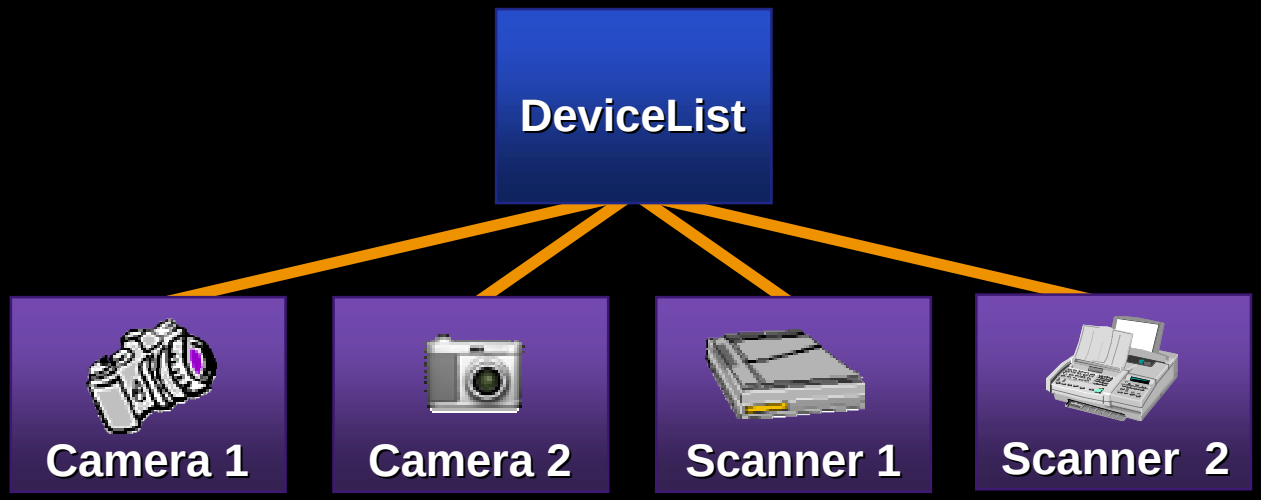
- Object-based API
  - C API
  - Object hierarchy
- Device-independent
  - Abstract to support various devices with the same API
    - Cameras
    - Scanners
- Transport-independent
  - USB
  - FireWire

Application Level APIs

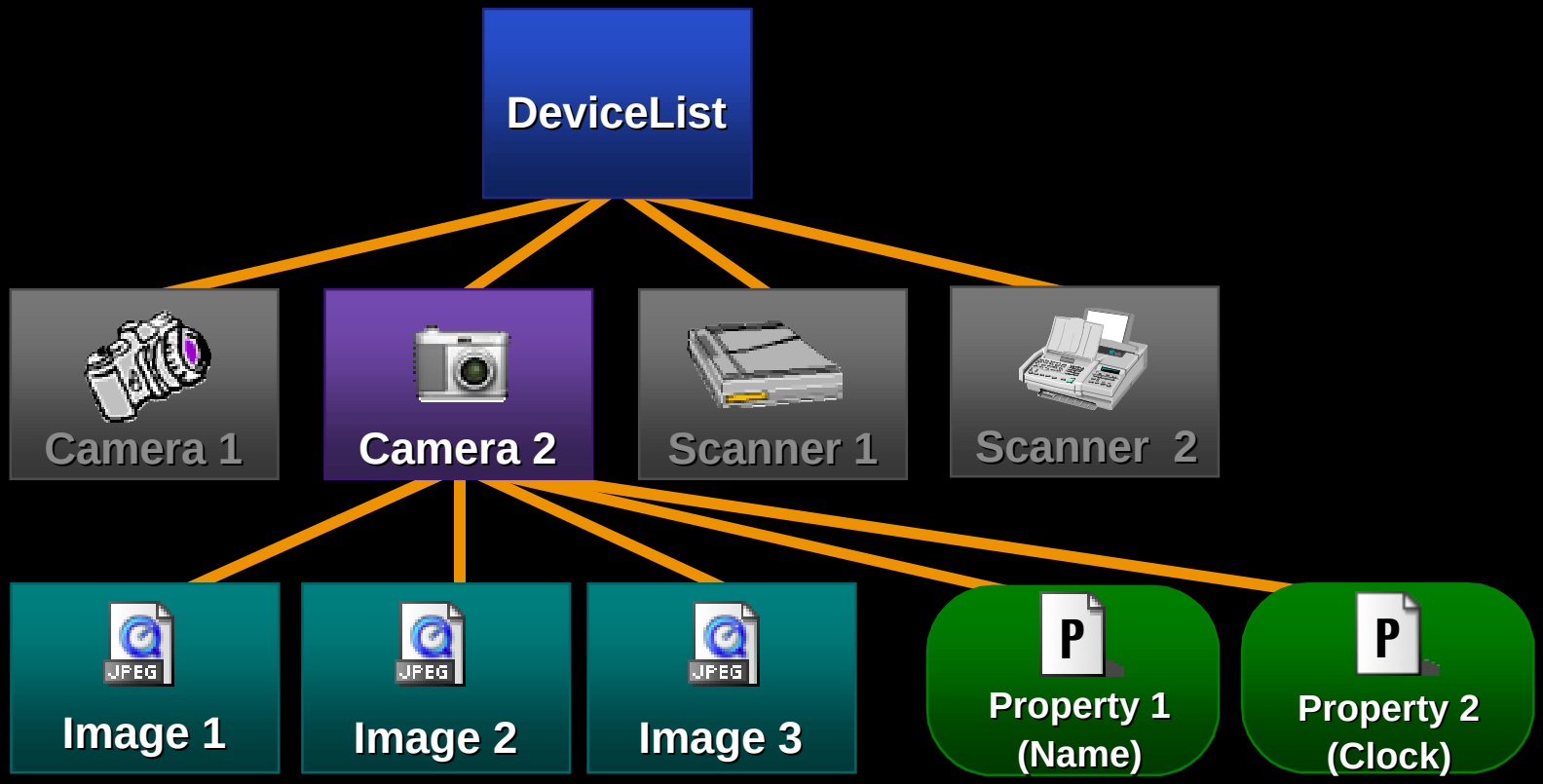
- Objects
  - Type and subtype as identifiers
  - Contain properties, and can also contain other objects
- Properties
  - Type and subtype as identifiers
  - Contain the “real” data



Object-based API



Object-based API



Object-based API

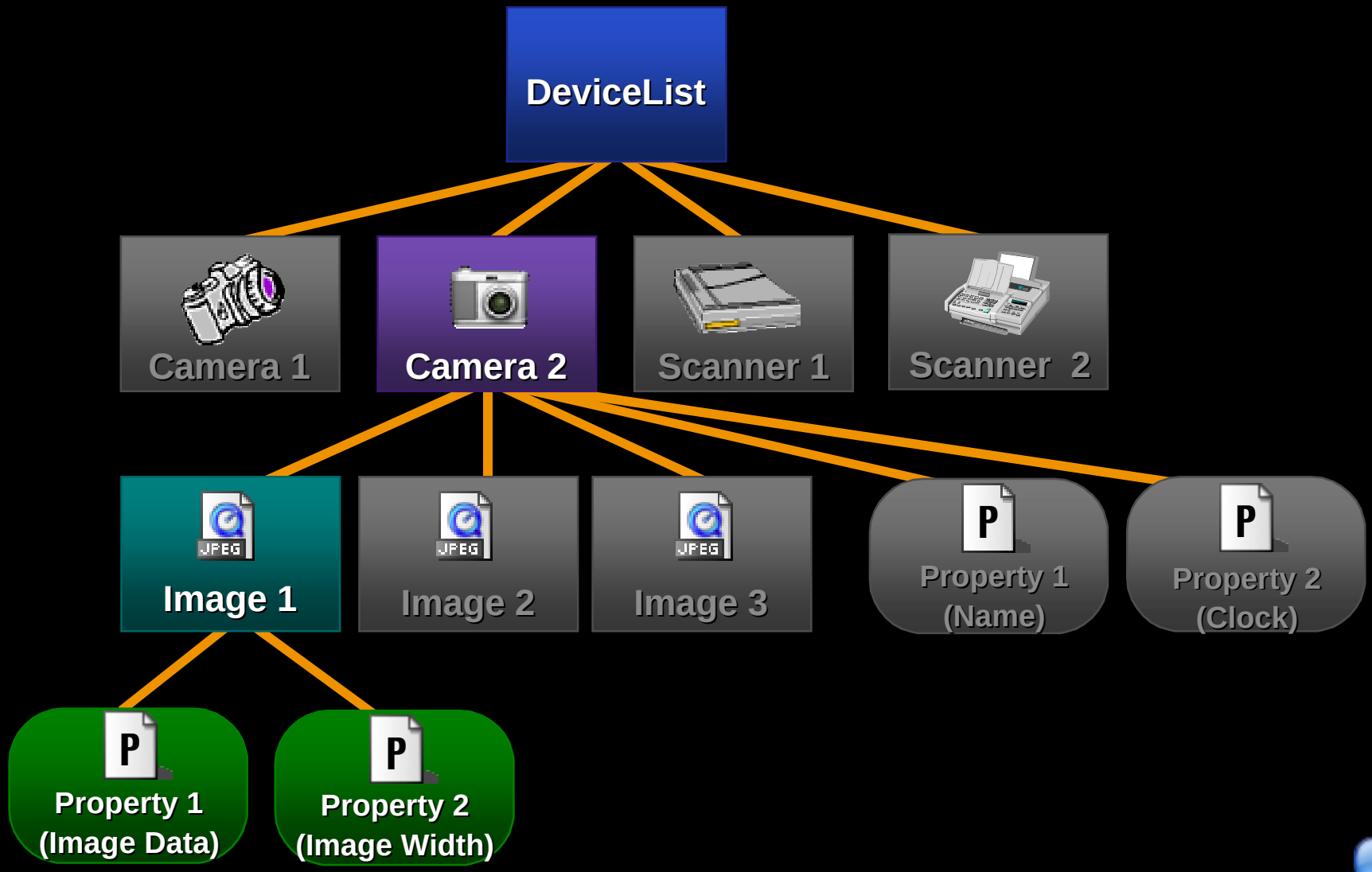


Image Capture API Overview

- Applications access the…
  - Device-objects via the device list object
  - Image-objects via the device-objects
  - Image-properties via the image-objects
  - Image data via the image-properties

Image Capture API Overview

- All functions have two parameters
  - An ICAxxxPB parameter block and
  - A completion routine
- Clear, and fill in appropriate elements before calling the ICA function, then extract elements afterwards

Image Capture API Overview

- General API structure

```
OSErr ICAxxx (ICAxxxPB *pb,
              ICACompletion completion  );
```

Image Capture API Overview

- All ICA functions can be called either synchronously or asynchronously
- Synchronous:
  - Pass nil as **completionProc**
- Asynchronous:
  - Pass your own function as **completionProc**, the API will return immediately, execute asynchronously, and call your completionProc when finished

Image Capture API Overview

- CompletionProc
  - Has a single parameter (ICAHeader) that needs to be casted to the appropriate ICAxxxPB parameter block

```
void YourCompletionProc(ICAHeader * header);
```

Image Capture API Overview

- Basic functions

ICAGetDeviceList
ICAObjectSendMessage
ICARegisterEventNotification

Image Capture API Overview

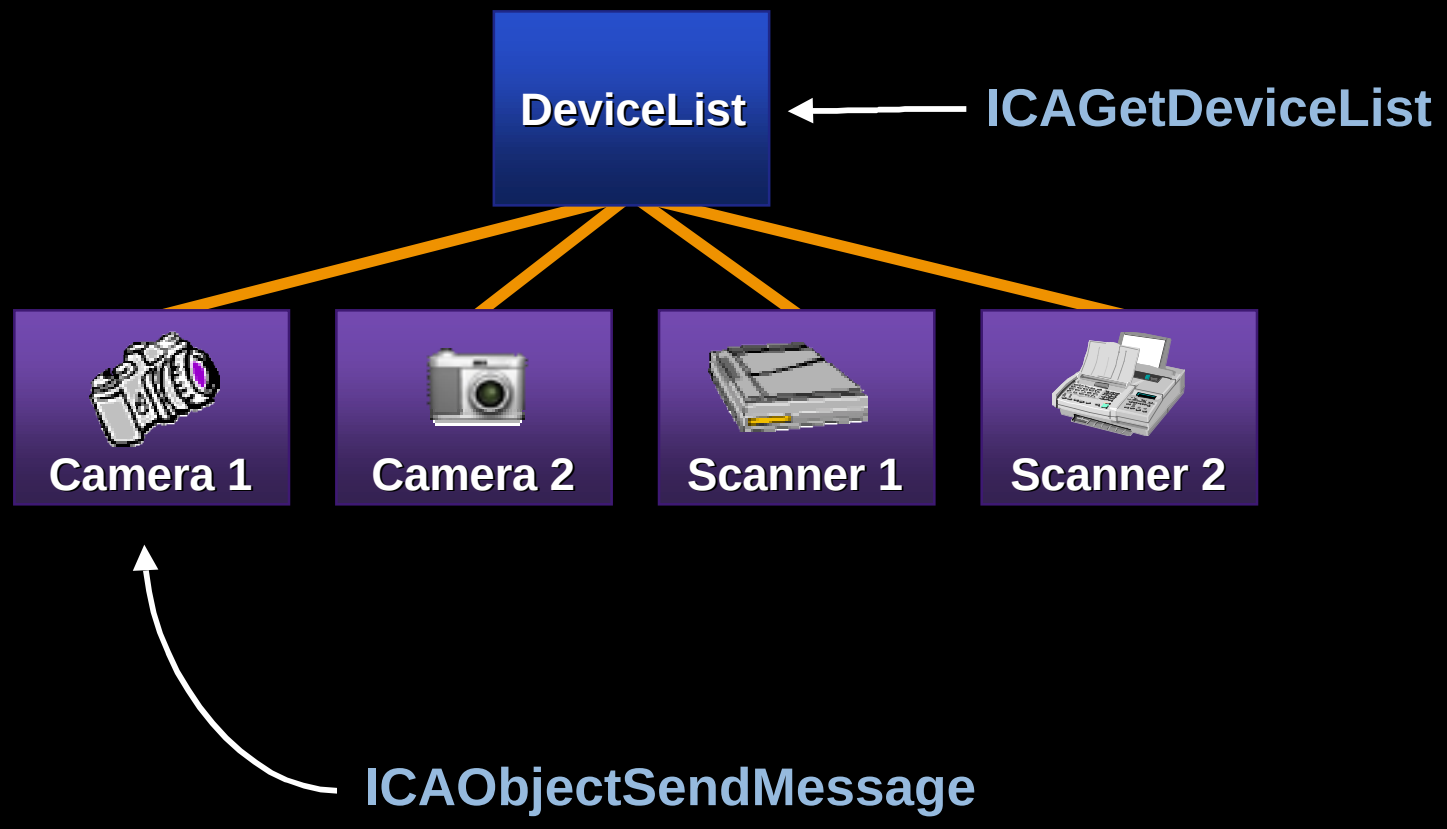
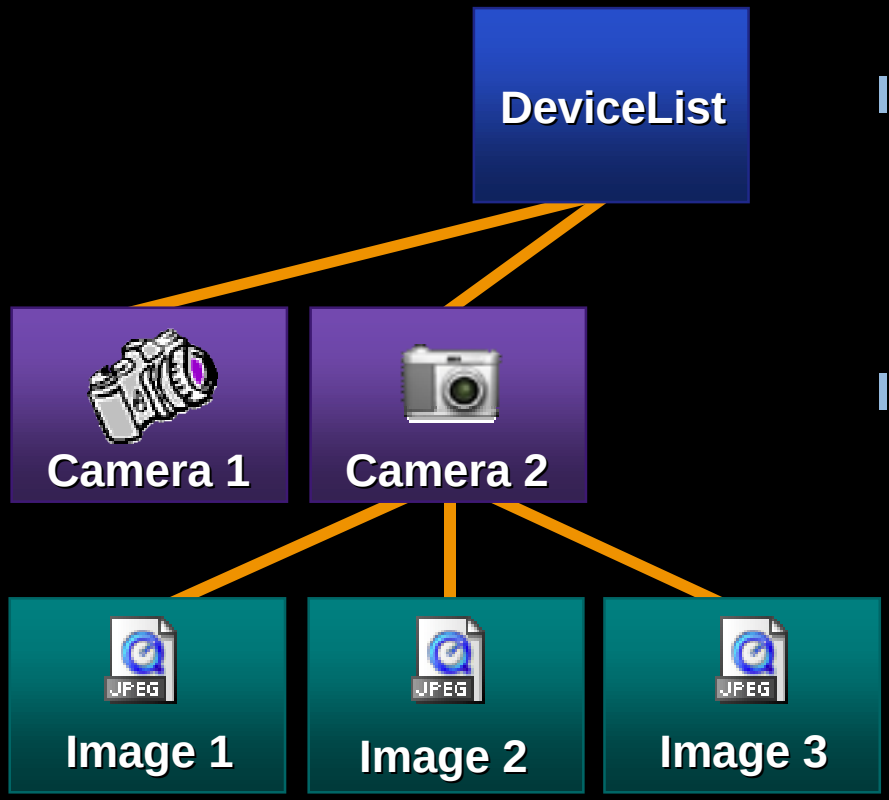


Image Capture API Overview

- Object related functions

ICAGetChildCount
ICAGetNthChild
ICAGetObjectInfo
ICAGetParentOfObject
ICAGetRootOfObject
ICAGetObjectRefCon
ICASetObjectRefCon

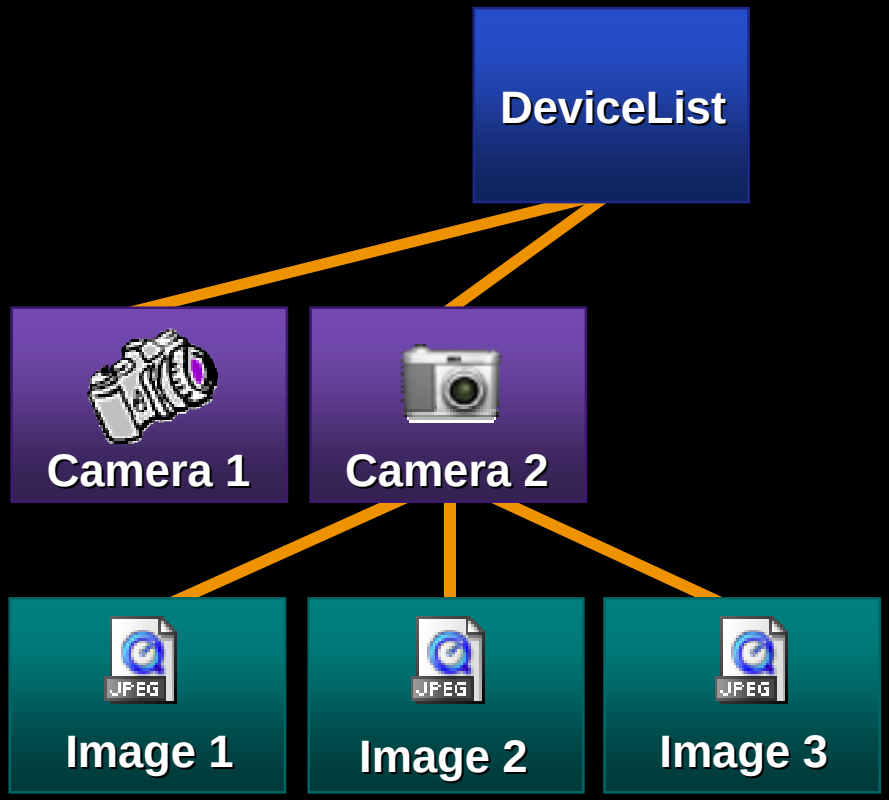
Image Capture API Overview



ICAGetChildCount 2

ICAGetChildCount 3

Image Capture API Overview



ICAGetObjectInfo

kICADevice

ICAGetObjectInfo

kICADeviceCamera

ICAGetObjectInfo

kICAFileImage

Image Capture API Overview

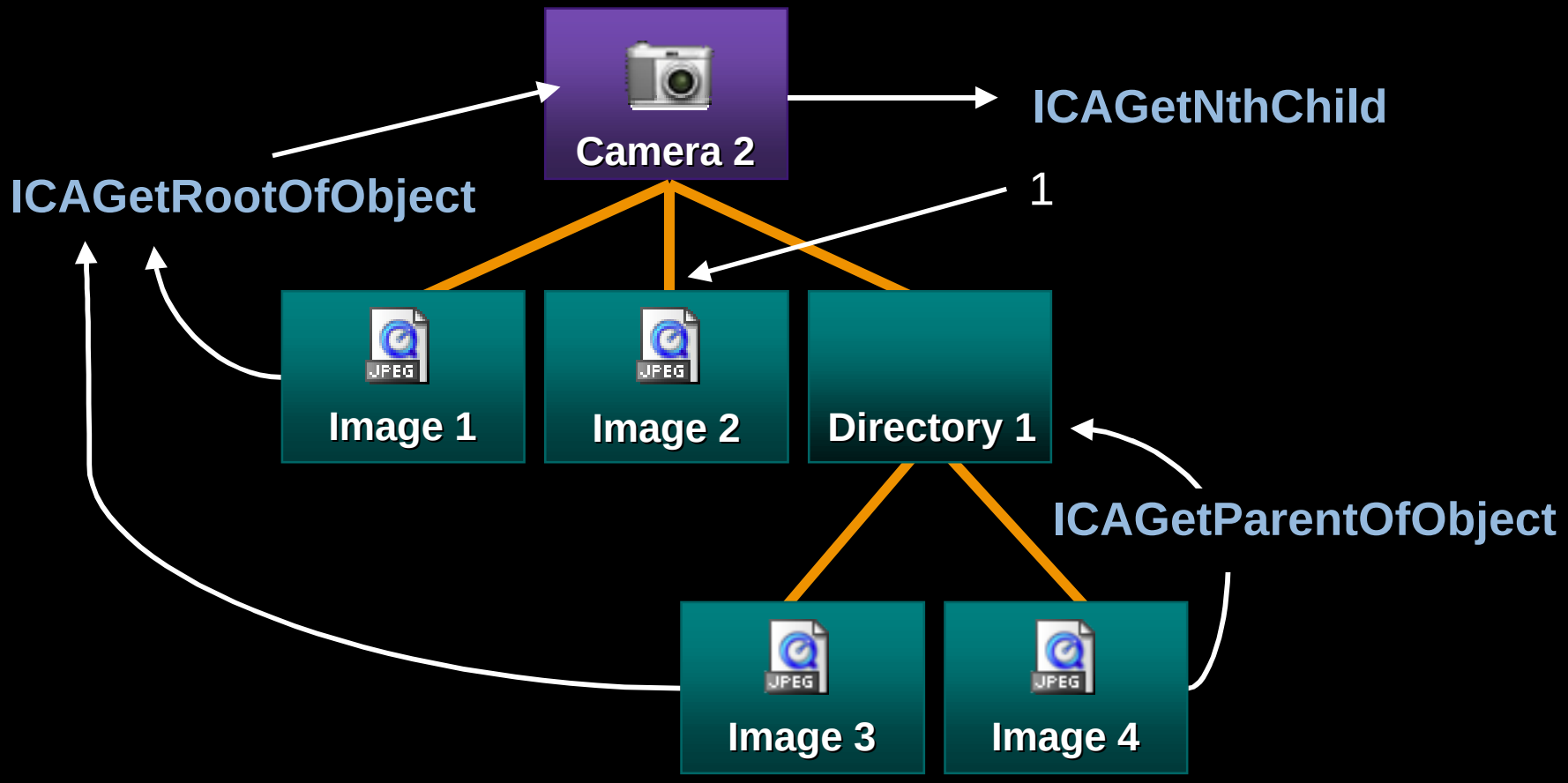


Image Capture API Overview

ICASetObjectRefCon

value

ICAGetObjectRefCon

value

Image Capture API Overview

- Property related functions

ICAGetPropertyCount
ICAGetNthProperty
ICAGetPropertyByType
ICAGetPropertyInfo
ICAGetPropertyData
ICASetPropertyData
ICAGetParentOfProperty
ICAGetRootOfProperty
ICAGetPropertyRefCon
ICASetPropertyRefCon

Image Capture API Overview

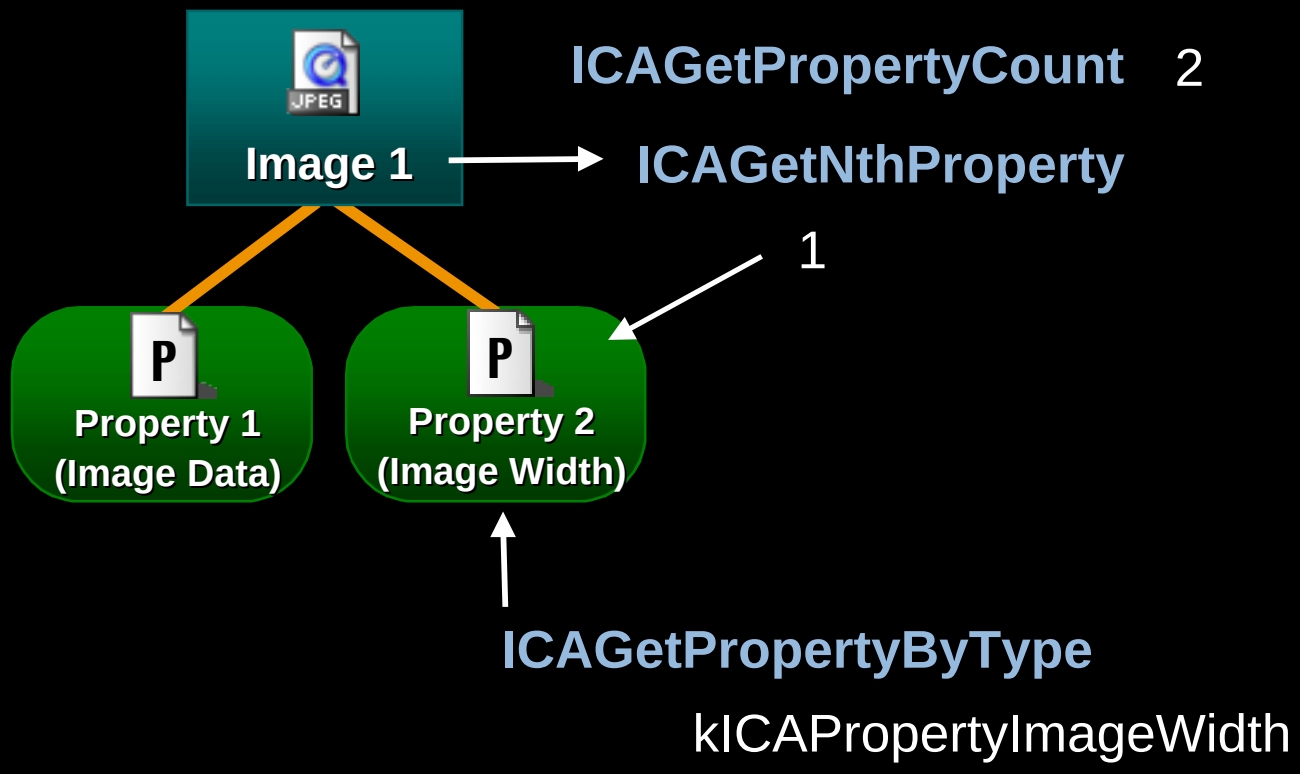


Image Capture API Overview

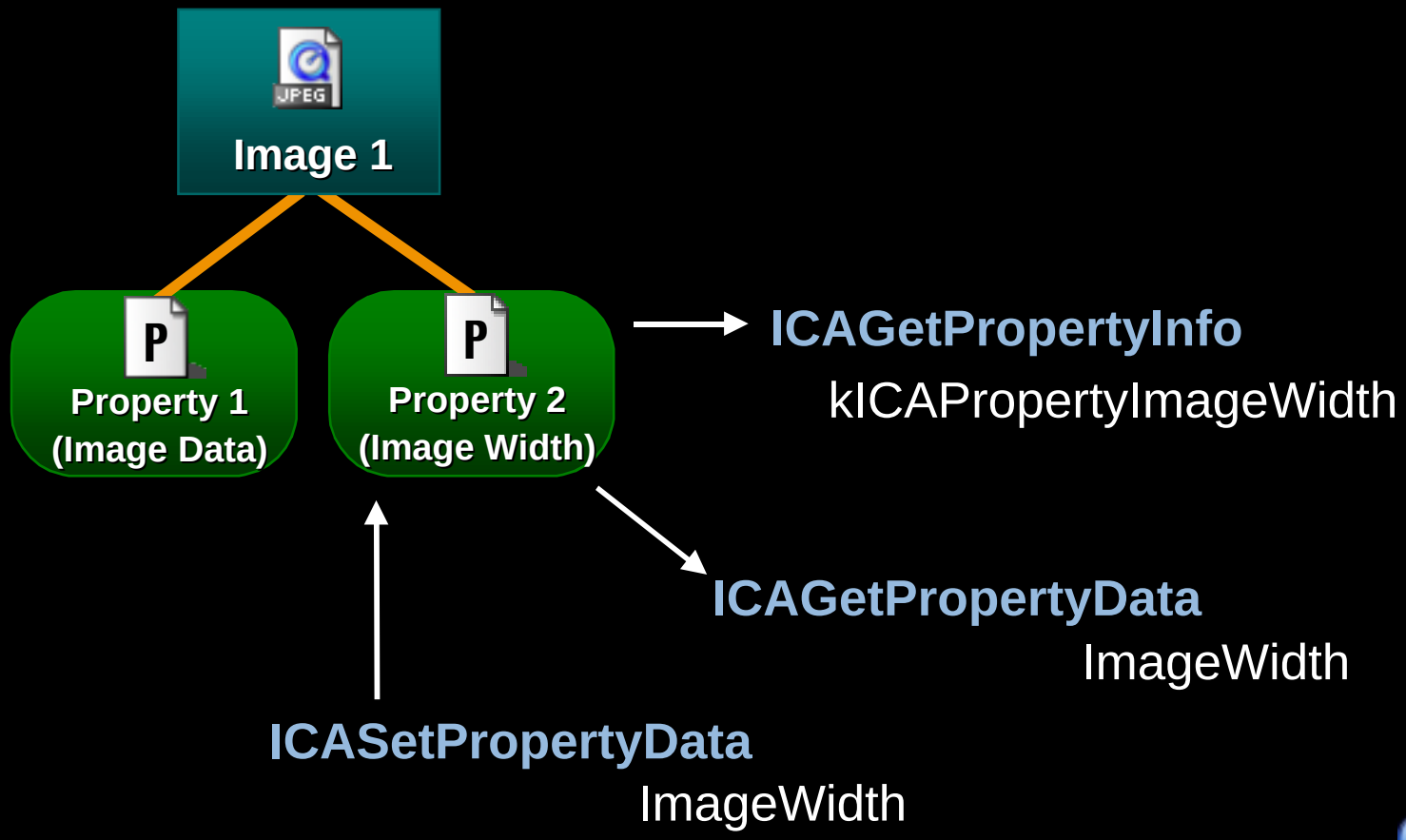


Image Capture API Overview

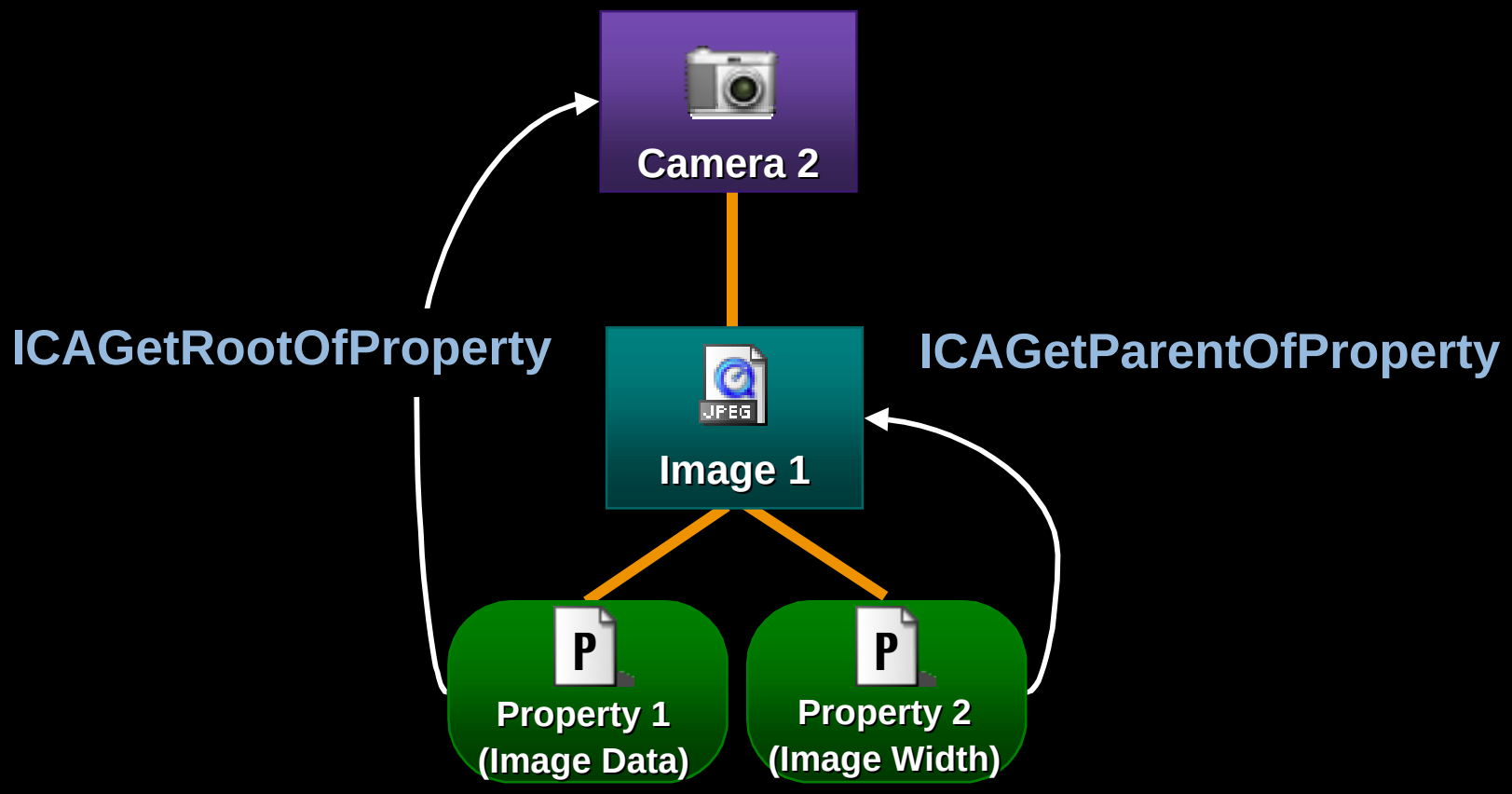
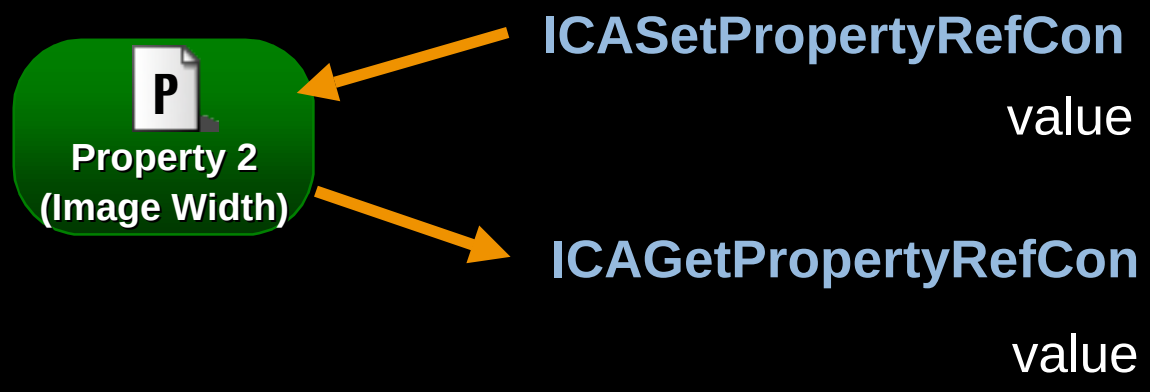


Image Capture API Overview



Demo

Image Capture Browser Tool and Sample Code

Session Schedule

- Introduction
- Architecture Overview
- Application Level APIs
- Device Level Support

Device Level APIs

- Camera modules are applications in user space
- Apple provides a camera module framework
- You have to implement 12 callback functions

Device Level APIs

- To create a camera module, you'll need the Image Capture SDK
- The Image Capture SDK will be available shortly from the ADC web site (www.apple.com/developer)

Summary

Image Capture advantages

- Great out of box experience for imaging devices
- Easy for developers to implement
- Device and Transport agnostic
- Minimum effort to add new device support
- Any application can become part of Apple's Digital Hub strategy

For More Information

- Photographic and Imaging Manufacturers Assoc.
  http://www.pima.net/
- USB Implementers Forum
  http://www.usb.org/
- Apple Developer Relations
  http://www.apple.com/developer/
- Other places
  - **www.apple.com/colorsync**
  - **www.apple.com/developer/sdk/**
  - **colorsync-dev and colorsync-users lists**
  - **colorsync@apple.com**

Roadmap

118 ColorSync: The color engine of Mac OS X	Room C **Wed., 2:00pm**
113 Text on Mac OS X: All about DrawThemeText, Unicode Edit Text Control and MLTE	Hall 2 **Wed., 3:30pm**
404 OpenGL: High Performance 2D Blazingly fast 2D graphics using OpenGL	Room J2 **Wed., 3:30pm**
124 Printing: Using Mac OS X's print architecture	Room A2 **Thurs., 3:30pm**

Roadmap

132 Graphics & Imaging Tips & Tricks: Optimize your graphics and printing	Room A2 **Fri., 9:00am**
FF118 Feedback Forum: Graphics Give us your feedback on Mac OS X graphics	Room J1 **Fri., 2:00pm**

Who to Contact

Travis W. Brown
Mac OS X Imaging Technology Manager
Apple Worldwide Developer Relations
travisb@apple.com

Q&A

Travis Brown
Mac OS X Imaging Technology Manager
travisb@apple.com

http://developer.apple.com/events/urls.html

WWDC

Worldwide Developers Conference 2001